A unifying approach to software and hardware design for scientific calculations and idempotent mathematics *

G. L. Litvinov (`litvinov@islc.msk.su`)
International Sophus Lie Centre

V. P. Maslov (`maslov@ipmnet.ru`)
Moscow State Institute of Electronics and Mathematics

A. Ya. Rodionov (`bs-rt@east.ru`)

Abstract. A unifying approach to software and hardware design generated by ideas of Idempotent Mathematics is discussed. The so-called idempotent correspondence principle for algorithms, programs and hardware units is described. A software project based on this approach is presented.

Keywords: universal algorithms, idempotent calculus, software design, hardware design, object oriented programming

1. Introduction

Numerical computations are still very important in computer applications. But until recently there has been a discrepancy between numerical methods and software/hardware tools for scientific calculations. In particular, numerical programming was not much influenced by the progress in Mathematics, programming languages and technology. Modern tools for numerical calculations are not unified, standardized and reliable enough. It is difficult to ensure the necessary accuracy and safety of calculations without loss of the efficiency and speed of data processing. It is difficult to get correct and exact estimations of calculation errors. For example, standard methods of interval arithmetic [2] do not allow to take into account error auto-correction effects [19] and, as a result, to estimate calculation errors accurately.

However, new ideas in Mathematics and Computer Science lead to a very promising approach (initially presented in [20]–[22]). An essential aspect of this approach is developing a system of algorithms, utilities and programs based on a new mathematical calculus which is called *Idempotent Analysis*, *Idempotent Calculus*, or *Idempotent Mathematics*. For many problems in optimization and mathematical modeling

---

this Idempotent Analysis plays the same unifying role as Functional Analysis in Mathematical Physics, see, e.g., [14], [17], [29]–[31] and surveys [15], [21].

Idempotent Analysis is based on replacing the usual arithmetic operations by a new set of basic operations (such as maximum or minimum). There is a lot of such new arithmetics, which are associated with sufficiently rich algebraic structures called idempotent semirings. It is very important that many problems that are nonlinear in the usual sense become linear with respect to an appropriate new arithmetic, i.e., linear over a suitable semiring (the so-called *idempotent superposition principle* [27], [28], [17], which is a natural analog of the well-known superposition principle in Quantum Mechanics). This 'linearity' considerably simplifies explicit construction of their solutions. Examples are the Bellman equation and its generalizations, the Hamilton–Jacobi equation, etc. The idempotent analysis serves as a powerful heuristic tool to construct new algorithms and apply unexpected analogies and ideas borrowed, e.g., from mathematical physics and quantum mechanics.

The abstract theory is well advanced and includes, in particular, a new integration theory, linear algebra and spectral theory, idempotent functional analysis, idempotent Fourier transforms, etc. Its applications include various optimization problems such as multi-criteria decision making, optimization on graphs, discrete optimization with a large parameter (asymptotic problems), optimal design of computer systems and computer media, optimal organization of parallel data processing, dynamic programming, applications to differential equations, numerical analysis, discrete event systems, computer science, discrete mathematics, mathematical logic, etc. (see, e.g., [1], [3], [5]–[15], [17], [21], [27]–[33] and references therein).

It is possible to implement this new approach to scientific and numeric calculations in the form of a powerful software system based on unified algorithms. This approach ensures the arbitrary necessary accuracy and safety of numerical calculations and working time reduction for developers and testers of algorithms because of software unification.

Our approach uses techniques of object oriented and functional programming (see, e.g., [26], [16]), which are very convenient for the design of our (suggested) software system. Computer algebra techniques [4] are also used. Modern techniques of systolic processors and VLSI realizations of numerical algorithms including parallel algorithms of linear algebra (see, e.g., [18], [32]) are convenient for effective implementations of the proposed approach to hardware design.

There is a regular method based on the idempotent theory for constructing back-end processors and technical devices intended for real-

ization of basic algorithms of idempotent calculus and mathematics of semirings. These hardware facilities can increase the speed of data processing.

2. Mathematical objects and their computer representations

Numerical algorithms are combinations of basic operations. Usually these basic operations deal with 'numbers'. In fact these 'numbers' are thought of as members of some numerical *domains* (real numbers, integers etc.). But every computer calculation deals with finite *models* (or finite *computer representations*) of these numerical domains. For example, integers can be modeled by integers modulo 2^n, real numbers can be represented by rational numbers or floating-point numbers etc. Discrepancies between mathematical objects (e.g., 'ideal' numbers) and their computer models (representations) lead to calculation errors.

Due to imprecision of sources of input data in real-world problems, the data usually come in the form of confidence intervals or other number sets rather than exact quantities. Interval Analysis (see, e.g., [2]) extends operations of traditional calculus from numbers to number intervals to make possible processing such imprecise data and controlling rounding errors in computational mathematics.

However, there is no universal model that would be good in all cases, so we have to use varieties of computer models. For example, real numbers can be represented by the following computer models:

– standard floating-point numbers,

– double precision floating-point numbers,

– arbitrary precision floating-point numbers,

– rational numbers,

– finite precision rational numbers,

– floating-slash and fixed-slash rational numbers,

– interval numbers,

– and others.

When examining an algorithm it is often useful to have a possibility to change computer representations of input/output data. For this aim the corresponding algorithm (and its software implementation) must be universal enough.

3. Universal algorithms

It is very important that many algorithms do not depend on particular models of a numerical domain and even on the domain itself. Algorithms of linear algebra (matrix multiplication, Gauss elimination etc.) are good examples of algorithms of this type.

Of course, one algorithm may be more universal than another algorithm solving the same problem. For example, numerical integration algorithms based on the Gauss–Jacobi quadrature formulas actually depend on computer models because they use finite precision constants. On the contrary, the rectangular formula and the trapezoid rule do not depend on models and in principle can be used even in the case of idempotent integration (see below).

The so-called object oriented software tools and programming languages (like C^{++} and Java, see, e.g., [26]) are very convenient for computer implementation of universal algorithms.

In fact there are no reasons to restrict ourselves to numerical domains only. Actually it may be a ring of polynomials, a field of rational functions, or an idempotent semiring. The case of idempotent semirings is extremely important because of numerous applications.

4. Idempotent correspondence principle

There is a nontrivial analogy between Mathematics of semirings and Quantum Mechanics. For example, the field of real numbers can be treated as a 'quantum object' with respect to idempotent semirings, which in turn can be treated as 'classical' or 'semi-classical' objects with respect to the former.

Let $\mathbb{R}$ be the field of real numbers and $\mathbb{R}_+$ the subset of all non-negative numbers. Consider the following change of variables:

$$u \mapsto w = h \ln u,$$

where $u \in \mathbb{R}_+ \setminus \{0\}$, $h > 0$; thus $u = e^{w/h}$, $w \in \mathbb{R}$. Denote by $\mathbf{0}$ the additional element $-\infty$ and by S the extended real line $\mathbb{R} \cup \{\mathbf{0}\}$. The above change of variables has a natural extension D_h to the whole S by $D_h(0) = \mathbf{0}$; also, we denote $D_h(1) = 0 = \mathbf{1}$.

Denote by S_h the set S equipped with the two operations $\oplus_h$ (generalized addition) and $\odot_h$ (generalized multiplication) such that D_h is a homomorphism of $\{\mathbb{R}_+, +, \cdot\}$ to $\{S, \oplus_h, \odot_h\}$. This means that $D_h(u_1 + u_2) = D_h(u_1) \oplus_h D_h(u_2)$ and $D_h(u_1 \cdot u_2) = D_h(u_1) \odot_h D_h(u_2)$, i.e., $w_1 \odot_h w_2 = w_1 + w_2$ and $w_1 \oplus_h w_2 = h \ln(e^{w_1/h} + e^{w_2/h})$. It is easy to prove that $w_1 \oplus_h w_2 \to \max\{w_1, w_2\}$ as $h \to 0$.

Denote by $\mathbb{R}_{\max}$ the set $S = \mathbb{R} \cup \{\mathbf{0}\}$ equipped with operations $\oplus = \max$ and $\odot = +$, where $\mathbf{0} = -\infty$, $\mathbf{1} = 0$ as above. Algebraic structures in $\mathbb{R}_+$ and S_h are isomorphic; therefore $\mathbb{R}_{\max}$ is a result of a deformation of the structure in $\mathbb{R}_+$.

We stress the obvious analogy with the quantization procedure, where h is the analog of the Planck constant. In these terms, $\mathbb{R}_+$ (or $\mathbb{R}$) plays the part of a 'quantum object' while $\mathbb{R}_{\max}$ acts as a 'classical' or 'semi-classical' object that arises as the result of a *dequantization* of this quantum object.

Likewise, denote by $\mathbb{R}_{\min}$ the set $\mathbb{R} \cup \{\mathbf{0}\}$ equipped with operations $\oplus = \min$ and $\odot = +$, where $\mathbf{0} = +\infty$ and $\mathbf{1} = 0$. Clearly, the corresponding dequantization procedure is generated by the change of variables $u \mapsto w = -h \ln u$.

Consider also the set $\mathbb{R} \cup \{\mathbf{0}, \mathbf{1}\}$, where $\mathbf{0} = -\infty$, $\mathbf{1} = +\infty$, together with the operations $\oplus = \max$ and $\odot = \min$. Obviously, it can be obtained as a result of a 'second dequantization' of $\mathbb{R}$ or $\mathbb{R}_+$.

The algebras presented in this section are the most important examples of idempotent semirings, the central algebraic structure of Idempotent Analysis.

Consider a set S equipped with two algebraic operations: *addition* $\oplus$ and *multiplication* $\odot$. The triple $\{S, \oplus, \odot\}$ is an *idempotent semiring* if it satisfies the following conditions (here and below, the symbol $\star$ denotes any of the two operations $\oplus$, $\odot$):

- the addition $\oplus$ and the multiplication $\odot$ are associative: $x \star (y \star z) = (x \star y) \star z$ for all $x, y, z \in S$;

- the addition $\oplus$ is commutative: $x \oplus y = y \oplus x$ for all $x, y \in S$;

- the addition $\oplus$ is *idempotent*: $x \oplus x = x$ for all $x \in S$;

- the multiplication $\odot$ is *distributive* with respect to the addition $\oplus$: $x \odot (y \oplus z) = (x \odot y) \oplus (x \odot z)$ and $(x \oplus y) \odot z = (x \odot z) \oplus (y \odot z)$ for all $x, y, z \in S$.

A *unity* of a semiring S is an element $\mathbf{1} \in S$ such that for all $x \in S$

$$\mathbf{1} \odot x = x \odot \mathbf{1} = x.$$

A *zero* of a semiring S is an element $\mathbf{0} \in S$ such that $\mathbf{0} \neq \mathbf{1}$ and for all $x \in S$

$$\mathbf{0} \oplus x = x, \qquad \mathbf{0} \odot x = x \odot \mathbf{0} = \mathbf{0}.$$

A semiring S is said to be *commutative* if $x \odot y = y \odot x$ for all $x, y \in S$. A commutative semiring is called a *semifield* if every nonzero

element of this semiring is invertible. It is clear that $\mathbb{R}_{\max}$ and $\mathbb{R}_{\min}$ are semifields.

Note that different versions of this axiomatics are used, see, e.g., [1], [3], [5], [6], [12], [13]–[15], [17], [21], [23], [31] and some literature indicated in these books and papers. Many nontrivial examples of idempotent semirings can be found, e.g., in [1], [5], [6], [12], [13], [14], [17], [21], [23], [25], [31]. For example, every vector lattice or ordered group can be treated as an idempotent semifield.

The addition $\oplus$ defines the following *standard partial order* on S: $x \preceq y$ if and only if $x \oplus y = y$. If S contains a zero $\mathbf{0}$, then $\mathbf{0} \preceq x$ for all $x \in S$. The operations $\oplus$ and $\odot$ are consistent with this order in the following sense: if $x \preceq y$, then $x \star z \preceq y \star z$ and $z \star x \preceq z \star y$ for all $x, y, z \in S$.

The basic object of the traditional calculus is a *function* defined on some set X and taking its values in the field $\mathbb{R}$ (or $\mathbb{C}$); its idempotent analog is a map $X \to S$, where X is some set and $S = \mathbb{R}_{\min}$, $\mathbb{R}_{\max}$, or another idempotent semiring. Let us show that redefinition of basic constructions of traditional calculus in terms of Idempotent Mathematics can yield interesting and nontrivial results (see, e.g., [17], [21], [23], [25], for details, additional examples and generalizations).

EXAMPLE 1. INTEGRATION AND MEASURES. To define an idempotent analog of the Riemann integral, consider a Riemann sum for a function $\varphi(x)$, $x \in X = [a, b]$, and substitute semiring operations $\oplus$ and $\odot$ for operations $+$ and $\cdot$ (usual addition and multiplication) in its expression (for the sake of being definite, consider the semiring $\mathbb{R}_{\max}$):

$$\sum_{i=1}^{N} \varphi(x_i) \cdot \Delta_i \quad \mapsto \quad \bigoplus_{i=1}^{N} \varphi(x_i) \odot \Delta_i = \max_{i=1,\ldots,N} \left(\varphi(x_i) + \Delta_i \right),$$

where $a = x_0 < x_1 < \cdots < x_N = b$, $\Delta_i = x_i - x_{i-1}$, $i = 1, \ldots, N$. As $\max_i \Delta_i \to 0$, the integral sum tends to

$$\int_X^{\oplus} \varphi(x)\, dx = \sup_{x \in X} \varphi(x)$$

for any function $\varphi \colon X \to \mathbb{R}_{\max}$ that is bounded. In general, for any set X the set function

$$m_\varphi(B) = \sup_{x \in B} \varphi(x), \quad B \subset X,$$

is called an $\mathbb{R}_{\max}$-*measure* on X; since $m_\varphi(\bigcup_\alpha B_\alpha) = \sup_\alpha m_\varphi(B_\alpha)$, this measure is completely additive. An idempotent integral with respect to this measure is defined as

$$\int_X^{\oplus} \psi(x)\, dm_\varphi = \int_X^{\oplus} \psi(x) \odot \varphi(x)\, dx = \sup_{x \in X} \left(\psi(x) + \varphi(x) \right).$$

Using the standard partial order it is possible to generalize these definitions for the case of arbitrary idempotent semirings.

EXAMPLE 2. FOURIER–LEGENDRE TRANSFORM. Consider the topological group $G = \mathbb{R}^n$. The usual Fourier–Laplace transform is defined as

$$\varphi(x) \mapsto \widetilde{\varphi}(\xi) = \int_G e^{i\xi \cdot x} \varphi(x)\, dx,$$

where $\exp(i\xi \cdot x)$ is a *character* of the group G, i.e., a solution of the following functional equation:

$$f(x + y) = f(x)f(y).$$

The idempotent analog of this equation is

$$f(x + y) = f(x) \odot f(y) = f(x) + f(y).$$

Hence 'idempotent characters' of the group G are linear functions of the form $x \mapsto \xi \cdot x = \xi_1 x_1 + \cdots + \xi_n x_n$. Thus the Fourier–Laplace transform turns into

$$\varphi(x) \mapsto \widetilde{\varphi}(\xi) = \int_G^{\oplus} \xi \cdot x \odot \varphi(x)\, dx = \sup_{x \in G} (\xi \cdot x + \varphi(x)).$$

This is the well-known Legendre (or Fenchel) transform.

These examples suggest the following formulation of the idempotent correspondence principle [20], [21]:

> *There exists a heuristic correspondence between interesting, useful and important constructions and results over the field of real (or complex) numbers and similar constructions and results over idempotent semirings in the spirit of N. Bohr's correspondence principle in Quantum Mechanics.*

So Idempotent Mathematics can be treated as a 'classical shadow (or counterpart)' of the traditional Mathematics over fields.

In particular, an idempotent version of Interval Analysis can be constructed [25]. The idempotent interval arithmetic appear to be remarkably simpler than its traditional analog. For example, in the traditional interval arithmetic multiplication of intervals is not distributive with respect to interval addition, while idempotent interval arithmetics conserve distributivity. Idempotent interval arithmetics are useful for reliable computing.

5.   Idempotent linearity

Let S be a commutative idempotent semiring.

The following example of a noncommutative idempotent semiring is very important.

EXAMPLE 3. Let $\mathrm{Mat}_n(S)$ be a set of all S-valued matrices, i.e., coefficients of these matrices are elements of S. Define the sum $\oplus$ of matrices $A = \|a_{ij}\|$, $B = \|b_{ij}\| \in \mathrm{Mat}_n(S)$ as $A \oplus B = \|a_{ij} \oplus b_{ij}\| \in \mathrm{Mat}_n(S)$. The *product* of two matrices $A \in \mathrm{Mat}_n(S)$ and $B \in \mathrm{Mat}_n(S)$ is a matrix $AB \in \mathrm{Mat}_n(S)$ such that $AB = \|\bigoplus_{k=1}^{m} a_{ik} \odot b_{kj}\|$. The set $\mathrm{Mat}_n(S)$ of square matrices is an idempotent semiring with respect to these operations. If $\mathbf{0}$ is the zero of S, then the matrix $O = \|o_{ij}\|$, where $o_{ij} = \mathbf{0}$, is the zero of $\mathrm{Mat}_n(S)$; if $\mathbf{1}$ is the unity of S, then the matrix $E = \|\delta_{ij}\|$, where $\delta_{ij} = \mathbf{1}$ if $i = j$ and $\delta_{ij} = \mathbf{0}$ otherwise, is the unity of $\mathrm{Mat}_n(S)$.

Now we discuss an idempotent analog of a linear space. A set V is called a *semimodule* over S (or an S-semimodule) if it is equipped with an idempotent commutative associative addition operation $\oplus_V$ and a multiplication $\odot_V \colon S \times V \to V$ satisfying the following conditions: for all λ, $\mu \in S$, v, $w \in V$

- $(\lambda \odot \mu) \odot_V v = \lambda \odot_V (\mu \odot_V v)$;

- $\lambda \odot_V (v \oplus_V w) = (\lambda \odot_V v) \oplus_V (\lambda \odot_V w)$;

- $(\lambda \oplus \mu) \odot_V v = (\lambda \odot_V v) \oplus_V (\mu \odot_V v)$.

An S-semimodule V is called a *semimodule with zero* if $\mathbf{0} \in S$ and there exists a *zero* element $\mathbf{0}_V \in V$ such that for all $v \in V$, $\lambda \in S$

- $\mathbf{0}_V \oplus_V v = v$;

- $\lambda \odot_V \mathbf{0}_V = \mathbf{0} \odot_V v = \mathbf{0}_V$.

EXAMPLE 4. FINITELY GENERATED FREE SEMIMODULE. The simplest S-semimodule is the direct product $S^n = \{(a_1, \ldots, a_n) \mid a_j \in S, j = 1, \ldots, n\}$. The set of all endomorphisms $S^n \to S^n$ coincides with the semiring $\mathrm{Mat}_n(S)$ of all S-valued matrices (see Example 3).

The theory of S-valued matrices, similar to the well-known Perron–Frobenius theory of nonnegative matrices, is well advanced and has very many applications, see, e.g., [1], [3], [5]–[15], [17], [21], [25], [30], [31]–[33]).

EXAMPLE 5. FUNCTION SPACES. An *idempotent function space* $\mathcal{F}(X; S)$ consists of functionals defined on a set X and taking their values in an idempotent semiring S. It is a subset of the set of all maps $X \to S$ such that if $f(x)$, $g(x) \in \mathcal{F}(X; S)$ and $c \in S$, then $(f \oplus g)(x) = f(x) \oplus g(x) \in \mathcal{F}(X; S)$ and $(c \odot f)(x) = c \odot f(x) \in \mathcal{F}(X; S)$; in other words, an idempotent function space is another example of an S-semimodule. If

the semiring S contains a zero element $\mathbf{0}$ and $\mathcal{F}(X; S)$ contains the zero constant function $o(x) = \mathbf{0}$, then the function space $\mathcal{F}(X; S)$ has the structure of a semimodule with the zero $o(x)$ over the semiring S. If the set X is finite we get the previous example.

Recall that the idempotent addition defines a standard partial order in S. An important example of an idempotent functional space is the space $\mathcal{B}(X; S)$ of all functions $X \to S$ bounded from above with respect to the partial order $\preceq$ in S. There are many interesting spaces of this type including $\mathcal{C}(X; S)$ (a space of continuous functions defined on a topological space X), analogs of the Sobolev spaces, etc. (see, e.g., [17], [21], [23], [29]–[31] for details).

According to the correspondence principle, many important concepts, ideas and results can be converted from usual Functional Analysis to Idempotent Analysis. For example, an idempotent scalar product in $\mathcal{B}(X; S)$ can be defined by the formula

$$\langle \varphi, \psi \rangle = \int_X^\oplus \varphi(x) \odot \psi(x)\, dx,$$

where the integral is defined as the 'sup' operation (see example 1).

EXAMPLE 6. INTEGRAL OPERATORS. It is natural to construct idempotent analogs of integral operators of the form

$$K : \varphi(y) \mapsto (K\varphi)(x) = \int_Y^\oplus K(x, y) \odot \varphi(y)\, dy,$$

where $\varphi(y)$ is an element of a functional space $\mathcal{F}_1(Y; S)$, $(K\varphi)(x)$ belongs to a space $\mathcal{F}_2(X; S)$ and $K(x, y)$ is a function $X \times Y \to S$. Such operators are *linear*, i.e., they are homomorphisms of the corresponding functional semimodules. If $S = \mathbb{R}_{\max}$, then this definition turns into the formula

$$(K\varphi)(x) = \sup_{y \in Y} (K(x, y) + \varphi(y)).$$

Formulas of this type are standard in optimization theory.

6. Superposition principle

In Quantum Mechanics the superposition principle means that the Schrödinger equation (which is basic for the theory) is linear. Similarly in Idempotent Mathematics the idempotent superposition principle means that some important and basic problems and equations (e.g., optimization problems, the Bellman equation and its versions and generalizations, the Hamilton-Jacobi equation) that are nonlinear

in the usual sense can be treated as linear over appropriate idempotent semirings, see [27]–[31], [17].

Linearity of the Hamilton-Jacobi equation over $\mathbb{R}_{\min}$ (and $\mathbb{R}_{\max}$) can be deduced from the usual linearity (over $\mathbb{C}$) of the corresponding Schrödinger equation by means of the dequantization procedure described above (in Section 4). In this case the parameter h of this dequantization coincides with $i\hbar$, where $\hbar$ is the Planck constant; so in this case $\hbar$ must take imaginary values (because $h > 0$; see [23] for details). Of course, this is closely related to variational principles of mechanics.

The situation is similar for the differential Bellman equation, see [17].

It is well-known that discrete versions of the Bellman equation can be treated as linear over appropriate idempotent semirings. The so-called *generalized stationary* (finite dimensional) *Bellman equation* has the form

$$X = AX \oplus B,$$

where X, A, B are matrices with elements from an idempotent semiring and the corresponding matrix operations are described in example 3 above; the matrices A and B are given (specified) and X is unknown.

B.A. Carré [5] used the idempotent linear algebra to show that different optimization problems for finite graphs can be formulated in unified manner and reduced to solving these Bellman equations, i.e., systems of linear algebraic equations over idempotent semirings. For example, Bellman's method of solving shortest path problems corresponds to a version of Jacobi's method for solving systems of linear equations, whereas Ford's algorithm corresponds to a version of Gauss-Seidel's method.

7. Correspondence principle for computations

Of course, the idempotent correspondence principle is valid for algorithms as well as for their software and hardware implementations [20]–[22]. Thus:

> *If we have an important and interesting numerical algorithm, then there is a good chance that its semiring analogs are important and interesting as well.*

In particular, according to the superposition principle, analogs of linear algebra algorithms are especially important. Note that numerical algorithms for standard infinite-dimensional linear problems over

idempotent semirings (i.e., for problems related to idempotent integration, integral operators and transformations, the Hamilton-Jacobi and generalized Bellman equations) deal with the corresponding finite-dimensional (or finite) 'linear approximations'. Nonlinear algorithms often can be approximated by linear ones. Thus the idempotent linear algebra is a basis for the idempotent numerical analysis.

Moreover, it is well-known that linear algebra algorithms are convenient for parallel computations; their idempotent analogs admit parallelization as well. Thus we obtain a systematic way of applying parallel computation to optimization problems.

Basic algorithms of linear algebra (such as inner product of two vectors, matrix addition and multiplication, etc.) often do not depend on concrete semirings, as well as on the nature of domains containing the elements of vectors and matrices. Algorithms to construct the closure $A^* = \mathbf{1} \oplus A \oplus A^2 \oplus \cdots \oplus A^n \oplus \cdots = \bigoplus_{n=1}^{\infty} A^n$ of an idempotent matrix A can be derived from standard methods for calculating $(\mathbf{1} - A)^{-1}$. For the Gauss–Jordan elimination method (via LU-decomposition) this trick was used in [32], and the corresponding algorithm is universal and can be applied both to the Bellman equation and to computing the inverse of a real (or complex) matrix $(\mathbf{1} - A)$. Computation of A^{-1} can be derived from this universal algorithm with some obvious cosmetic transformations.

Thus it seems reasonable to develop universal algorithms that can deal equally well with initial data of different domains sharing the same basic structure [21], [22].

8. Correspondence principle for hardware design

A systematic application of the correspondence principle to computer calculations leads to a unifying approach to software and hardware design.

The most important and standard numerical algorithms have many hardware realizations in the form of technical devices or special processors. *These devices often can be used as prototypes for new hardware units generated by substitution of the usual arithmetic operations for its semiring analogs and by adding a representation of neutral elements* $\mathbf{0}$ *and* $\mathbf{1}$ (the latter usually is not difficult). Of course, the case of numerical semirings consisting of real numbers (maybe except neutral elements) is the most simple and natural [20]–[22]. Note that for semifields (including $\mathbb{R}_{\max}$ and $\mathbb{R}_{\min}$) the operation of division is also defined.

Good and efficient technical ideas and decisions can be transposed from prototypes into new hardware units. Thus the correspondence principle generates a regular heuristic method for hardware design. Note that to get a patent it is necessary to present the so-called 'invention formula', that is to indicate a prototype for the suggested device and the difference between these devices.

Consider (as a typical example) the most popular and important algorithm of computing the scalar product of two vectors:

$$(x, y) = x_1 y_1 + x_2 y_2 + \ldots + x_n y_n. \tag{1}$$

The universal version of (1) for any semiring A is obvious:

$$(x, y) = (x_1 \odot y_1) \oplus (x_2 \odot y_2) \oplus \ldots \oplus (x_n \odot y_n). \tag{2}$$

In the case $A = \mathbb{R}_{\max}$ this formula turns into the following one:

$$(x, y) = \max\{x_1 + y_1, x_2 + y_2, \ldots, x_n + y_n\}. \tag{3}$$

This calculation is standard for many optimization algorithms, so it is useful to construct a hardware unit for computing (3). There are many different devices (and patents) for computing (1) and every such device can be used as a prototype to construct a new device for computing (3) and even (2). Many processors for matrix multiplication and for other algorithms of linear algebra are based on computing scalar products and on the corresponding 'elementary' devices respectively, etc.

There are some methods to make these new devices more universal than their prototypes. There is a modest collection of possible operations for standard numerical semirings: max, min, and the usual arithmetic operations. So, it is easy to construct programmable hardware processors with variable basic operations. Using modern technologies it is possible to construct cheap special-purpose multiprocessor chips implementing reliable, thoroughly tested algorithms. The so-called systolic processors are especially convenient for this purpose. A systolic array is a 'homogeneous' computing medium consisting of elementary processors, where the general scheme and processor connections are simple and regular. Every elementary processor pumps data in and out performing elementary operations in such a way that the corresponding data flow is kept up in the computing medium; there is an analogy with the blood circulation, hence the name 'systolic' (see, e.g., [18], [32]).

Of course, hardware implementations for important and popular basic algorithms can increase the speed of data processing.

9.   Correspondence principle for software design

Software implementations for universal semiring algorithms are not so efficient as hardware ones (with respect to the computation speed) but are much more flexible. Program modules can deal with abstract (and variable) operations and data types. Concrete values for these operations and data types can be defined by the corresponding input data. In this case concrete operations and data types are generated by means of additional program modules. For programs written in this manner it is convenient to use special techniques of the so-called object oriented (and functional) design, see, e.g., [26], [16]. Fortunately, powerful tools supporting the object-oriented software design have recently appeared including compilers for real and convenient programming languages (such as C^{++} and Java).

We propose a project to obtain an implementation of the correspondence principle approach to scientific calculations in the form of a powerful software system based on a collection of universal algorithms. This approach ensures working time reduction for programmers and users because of software unification. The arbitrary necessary accuracy and safety of numeric calculations can be ensured as well.

The system contains several levels (including programmer and user levels) and many modules.

Roughly speaking, it is divided into three parts. The first part contains modules that implement domain modules (finite representations of basic mathematical objects). The second part implements universal (invariant) calculation methods. The third part contains modules implementing model dependent algorithms. These modules may be used in user programs written in C^{++} and Java.

The following modules and algorithm implementations are in progress:

− Domain modules:

  - infinite precision integers;
  - rational numbers;
  - finite precision rational numbers;
  - finite precision complex rational numbers;
  - fixed- and floating-slash rational numbers;
  - complex rational numbers;
  - arbitrary precision floating-point real numbers;
  - arbitrary precision complex numbers;
  - p-adic numbers;

- interval numbers;
- ring of polynomials over different rings;
- idempotent semirings $R(\max, \min)$, $R(\max, +)$, $R(\min, +)$, interval idempotent semirings
- and others.

− Algorithms:

- linear algebra;
- numerical integration;
- roots of polynomials;
- spline interpolations and approximations;
- rational and polynomial interpolations and approximations;
- special functions calculation;
- differential equations;
- optimization and optimal control;
- and others.

This software system may be especially useful for designers of algorithms, software engineers, students and mathematicians.

References

1. A.V. Aho, J.E. Hopcroft and J.D. Ullman, *The Design and Analysis of Computer Algorithms* (Addison–Wesley, Reading Mass., 1976).
2. G. Alefeld and J. Herzberger, *Introduction to Interval Computations* (Academic Press, New York, 1983).
3. F. Baccelli, G. Cohen, G.J. Olsder and J.-P. Quadrat, *Synchronization and Linearity: An Algebra for Discrete Event Systems* (Wiley, New York, 1992).
4. B. Buchberger et al., eds., *Computer Algebra. Symbolic and Algebraic Computation* (Springer, Wien and New York, 1982).
5. B.A. Carré, An algebra for Network Routine Problems, *J. Inst. Math.* **7** (1971) 273–294.
6. B.A. Carré, *Graphs and Networks* (The Clarendon Press / Oxford University Press, Oxford, 1979).
7. J. Cochet-Terrasson, G. Cohen, S. Gaubert, M. McGettrick and J.-P. Quadrat, Numerical computation of spectral elements in max-plus algebra, in: Proc. of the IFAC Workshop on Systems Structure and Control (Nantes, July 1998).
8. G. Cohen, S. Gaubert, and J.-P. Quadrat, Max-plus algebra and system theory: where we are and where to go now, in: Proc. of the IFAC Workshop on Systems Structure and Control (Nantes, July 1998).
9. R.A. Cuninghame-Green, *Minimax Algebra*, Lecture Notes in Economics and Mathematical Systems, vol. 166 (Springer, Berlin, 1979).

10.   S. Gaubert, *Théorie des systèmes linéaires dans les dioïdes*, Ph. D. Thesis, École des Mines de Paris, 1992.

11.   S. Gaubert and M. Plus, Methods and applications of (max,+) linear algebra, in: R. Reischuk and M. Morvan, eds., Proc. STACS'97, *Lecture Notes in Computer Science*, vol. 500 (Springer, Berlin, 1997).

12.   J.S. Golan, *The Theory of Semirings with Applications in Mathematics and Theoretical Computer Science* (Longman Sci. and Techn., Harlow and New York, 1992).

13.   M. Gondran and M. Minoux, *Graphes et algorithmes* (Eyrolles, Paris, 1979).

14.   J. Gunawardena, ed., *Idempotency*, Publ. of the Newton Institute (Cambridge University Press, Cambridge, 1998).

15.   J. Gunawardena, An introduction to idempotency, in: J. Gunawardena, ed., *Idempotency*, Publ. of the Newton Institute (Cambridge University Press, Cambridge, 1998) 1–49.

16.   P. Henderson, *Functional Programming. Application and Implementation* (Prentice Hall, Englewood Cliffs–N.J., 1980).

17.   V.N. Kolokoltsov and V.P. Maslov, *Idempotent Analysis and Applications* (Kluwer Acad. Publ., Dordrecht, 1997).

18.   H.T. Kung, Two-lined pipelined systolic arrays for matrix multiplication, polynomial evaluation and discrete Fourier transformation, in: J. Demongeof et al., eds., *Dynamical Systems and Cellular Automata* (Academic Press, New York, 1985) 321–330.

19.   G.L. Litvinov, Approximate construction of rational approximations. Applications, *Russian Journal of Mathematical Physics* **1**, # **3** (1993) 313-352. Also available at `http://arXiv.org`, e-print math.NA/0101042.

20.   G.L. Litvinov and V.P. Maslov, Idempotent Mathematics: A Correspondence Principle and its Applications to Computing, *Russian Mathematical Surveys* **51**, # **6** (1996) 1210–1211.

21.   G. L. Litvinov and V. P. Maslov, Correspondence principle for idempotent calculus and some computer applications, in: J. Gunawardena, ed., *Idempotency*, Publ. of the Newton Institute (Cambridge University Press, Cambridge, 1998) 420–443; also Preprint IHES/M/95/33, Institut des Hautes Etudes Scientifiques, Bures-sur-Yvette, 1995. Also available at `http://arXiv.org`, e-print math.GM/0101021.

22.   G.L. Litvinov, V.P. Maslov and A.Ya. Rodionov, *Unifying Approach to Software and Hardware Design for Scientific Calculations*, Preprint, International Sophus Lie Centre, Moscow, 1995. Also available at `http://arXiv.org`, e-print quant-ph/9904024.

23.   G.L. Litvinov, V.P. Maslov and G.B. Shpiz, *Idempotent Functional Analysis. An Algebraic Approach*, Preprint, International Sophus Lie Centre, Moscow, 1998 (in Russian; to be published in English in Math. Notes). Also available at `http://arXiv.org`, e-print math.FA/0009128.

24.   G.L. Litvinov, V.P. Maslov and G.B. Shpiz, Linear Functionals on Idempotent Spaces: An Algebraic Approach, *Soviet Math. Dokl.* **58**, #**3** (1999) 389–391. Also available at `http://arXiv.org`, e-print math.FA/0012268.

25.   G.L. Litvinov, V.P. Maslov and A.N. Sobolevskiĭ, *Idempotent Mathematics and Interval Analysis*, Preprint ESI 632, The Erwin Schrödinger International Institute for Mathematical Physics, Vienna, 1998; e-print: `http://www.esi.ac.at` and `http://arXiv.org`, e-print math.SC/9911126.

26.   M. Lorenz (1993), *Object Oriented Software Development: A Practical Guide* (Prentice Hall, Englewood Cliffs–N. J., 1993).

27. V.P. Maslov, A New Approach to Generalized Solutions of Non-linear Systems, *Soviet Math. Dokl.* **42**, #**1** (1987) 29–33.

28. V.P. Maslov, New Superposition Principle for Optimization Problems, *Russian Math. Surveys* **42**, # **3** (1987) 43–54.

29. V.P. Maslov, *Méthodes opératorielles* (Editions MIR, Moscow, 1987).

30. V.P. Maslov, K.A. Volosov, eds., *Mathematical Aspects of Computer Engineering* (MIR Publ., Moscow, 1988).

31. V.P. Maslov and S.N. Samborskiĭ, eds., *Idempotent Analysis*, Advances in Soviet Mathematics, vol. 13 (American Mathematical Society, Providence R. I., 1992).

32. G. Rote, A Systolic Array Algorithm for the Algebraic Path Problem (Shortest Paths; Matrix Inversion) *Computing* **34** (1985) 191–219.

33. U. Zimmerman, Linear and Combinatorial Optimization in Ordered Algebraic Structures, *Ann. Discrete Math.* **10** (1981) 1–380.

Address for Offprints: INTERNATIONAL SOPHUS LIE CENTRE
Nagornaya, 27–4–72, Moscow 113186 Russia
e-mail: LITVINOV@ISLC.MSK.SU